We claim:

1. A method of upgrading a software application from a user modified prior version to an upgrade version, wherein the user modified prior version and the upgrade version have a common ancestor version, said method comprising:

determining a first set of differences based on a comparison of the user modified prior version and the common ancestor version;

determining a second set of differences based on a comparison of the upgrade version and the common ancestor version;

determining which differences from said first and second sets of differences are compatible differences and which are conflicting differences; and

applying changes to the upgrade version associated with said compatible differences.

2. The method of claim 1, wherein the user modified prior version comprises a first plurality of objects and the common ancestor version comprises a second plurality of objects, and wherein said determining said first set of differences comprises:

determining whether one or more objects from said first plurality of objects share a common name with one or more objects from said second plurality of objects, and if so, determining whether attributes associated with the commonly named objects are identical, and if not, including data related to the difference between said attributes in said first set of differences.

3. The method of claim 2, wherein the difference between said attributes is determined to be a compatible difference if said attributes are superficial.

4. The method of claim 3, wherein said attributes are designated as being superficial by a user.

5. The method of claim 2, wherein the upgrade version comprises a third plurality of objects, and wherein said determining said second set of differences comprises:

determining whether one or more objects from said second plurality of objects share a common name with one or more objects from said third plurality of objects, and if so, determining whether attributes associated with the commonly named objects are identical, and if not, including data related to the difference between said attributes in said second set of differences.

6. The method of claim 1, further comprising determining a third set of differences based on a comparison of the upgrade version and the user modified prior version.

7. The method of claim 6, wherein the user modified prior version comprises a first plurality of objects, the common ancestor version comprises a second plurality of objects, and the upgrade version comprises a third plurality of objects, and wherein said determining said third set of differences comprises:

determining whether an object from said first plurality of objects is not included within said second or third plurality of objects, and if so, indicating that the addition of said object to said first plurality of objects is a compatible difference.

8. The method of claim 7, wherein said applying changes to the upgrade version associated with said compatible differences comprises adding said object to the upgrade version.

9. The method of claim 6, wherein the user modified prior version comprises a first plurality of objects, the common ancestor version comprises a second plurality of objects, and the upgrade version comprises a third plurality of objects, and wherein said determining said third set of differences comprises:

determining whether an object from said second and third plurality of objects is not included within said first plurality of objects, and if so, indicating that the deletion of said object from said first plurality of objects is a conflicting difference.

10. The method of claim 9, wherein said applying changes to the upgrade version associated with said compatible differences comprises adding said object to the upgrade version.

11. Computer executable software code for upgrading a software application from a user modified prior version to an upgrade version, wherein the user modified prior

version and the upgrade version have a common ancestor version, said software code comprising:

code to determine a first set of differences based on a comparison of the user modified prior version and the common ancestor version;

code to determine a second set of differences based on a comparison of the upgrade version and the common ancestor version;

code to determine which differences from said first and second sets of differences are compatible differences and which are conflicting differences; and

code to apply changes to the upgrade version associated with said compatible differences.

12. The software code of claim 11, wherein the user modified prior version comprises a first plurality of objects and the common ancestor version comprises a second plurality of objects, and wherein said code to determine said first set of differences comprises:

code to determine whether one or more objects from said first plurality of objects share a common name with one or more objects from said second plurality of objects, and if so, to determine whether attributes associated with the commonly named objects are identical, and if not, to include data related to the difference between said attributes in said first set of differences.

13. The software code of claim 12, wherein the difference between said attributes is determined to be a compatible difference if said attributes are superficial.

14. The software code of claim 12, wherein the upgrade version comprises a third plurality of objects, and wherein said code to determine said second set of differences comprises:

code to determine whether one or more objects from said second plurality of objects share a common name with one or more objects from said third plurality of objects, and if so, to determine whether attributes associated with the commonly named objects are identical, and if not, to include data related to the difference between said attributes in said second set of differences.

15. The software code of claim 11, further comprising code to determine a third set of differences based on a comparison of the upgrade version and the user modified prior version.

16. The software code of claim 15, wherein the user modified prior version comprises a first plurality of objects, the common ancestor version comprises a second plurality of objects, and the upgrade version comprises a third plurality of objects, and wherein said code to determine said third set of differences comprises:

code to determine whether an object from said first plurality of objects is not included within said second or third plurality of objects, and if so, to indicate that the addition of said object to said first plurality of objects is a compatible difference.

17. The software code of claim 16, wherein said code to apply changes to the upgrade version associated with said compatible differences comprises code to add said object to the upgrade version.

18. The software code of claim 17, wherein the user modified prior version comprises a first plurality of objects, the common ancestor version comprises a second plurality of objects, and the upgrade version comprises a third plurality of objects, and wherein said code to determine said third set of differences comprises:

code to determine whether an object from said second and third plurality of objects is not included within said first plurality of objects, and if so, to indicate that the deletion of said object from said first plurality of objects is a conflicting difference.

19. The software code of claim 18, wherein said code to apply changes to the upgrade version associated with said compatible differences comprises code to add said object to the upgrade version.

20. An apparatus for upgrading a software application from a user modified prior version to an upgrade version, wherein the user modified prior version and the upgrade version have a common ancestor version, said apparatus comprising:

means for determining a first set of differences based on a comparison of the user modified prior version and the common ancestor version;

means for determining a second set of differences based on a comparison of the upgrade version and the common ancestor version;

means for determining which differences from said first and second sets of differences are compatible differences and which are conflicting differences; and

means for applying changes to the upgrade version associated with said compatible differences.

21. The apparatus of claim 20, wherein the user modified prior version comprises a first plurality of objects and the common ancestor version comprises a second plurality of objects, and wherein said means for determining said first set of differences comprises:

means for determining whether one or more objects from said first plurality of objects share a common name with one or more objects from said second plurality of objects, and if so, for determining whether attributes associated with the commonly named objects are identical, and if not, for including data related to the difference between said attributes in said first set of differences.

22. A method of upgrading a software application from a user modified prior version to an upgrade version, wherein the user modified prior version and the upgrade version have a common ancestor version, said method comprising:

comparing the user modified prior version, the common ancestor version, and the upgrade version to determine differences;

determining which of said differences are compatible and which are conflicting; and

applying changes to the upgrade version associated with said compatible differences.

23. The method of claim 22, wherein the user modified prior version comprises a first plurality of objects, the common ancestor version comprises a second plurality of objects, the upgrade version comprises a third plurality of objects, and wherein said comparing comprises comparing said first plurality of objects with said second and third pluralities of objects to determine whether objects were added, deleted, or modified by a user.

24. The method of claim 23, wherein said determining comprises:

indicating that differences associated with objects added by said user are compatible differences;

indicating that differences associated with objects deleted by said user are conflicting differences; and

indicating that differences associated with objects modified by said user are compatible differences if said objects modified by said user are superficial.

25. Computer executable software code for upgrading a software application from a user modified prior version to an upgrade version, wherein the user modified prior version and the upgrade version have a common ancestor version, said software code comprising:

code to compare the user modified prior version, the common ancestor version, and the upgrade version to determine differences;

code to determine which of said differences are compatible and which are conflicting; and

code to apply changes to the upgrade version associated with said compatible differences.

26. The software code of claim 25, wherein the user modified prior version comprises a first plurality of objects, the common ancestor version comprises a second plurality of objects, the upgrade version comprises a third plurality of objects, and wherein said code to compare comprises code to compare said first plurality of objects with said second and third pluralities of objects to determine whether objects were added, deleted, or modified by a user.

27. The software code of claim 26, wherein said code to determine comprises:

code to indicate that differences associated with objects added by said user are compatible differences;

code to indicate that differences associated with objects deleted by said user are conflicting differences; and code to indicate that differences associated with objects modified by said user are compatible differences if said objects modified by said user are superficial.